

5

10

15

Patent Application of

Haw-Minn Lu for

20

Width Upgrade for a Scalable Switching Network

25

Background - Field of Invention

30

This invention relates to redundant multistage switching networks, specifically to the non-stop addition of a new router to each row of such a network.

35

Background – Discussion of Prior Art

40

When a multistage switching network such as a Banyan or Butterfly network is expanded by width a duplicate copy of itself must be added. In addition, adding a new row requires that half of the external connections have to be disconnected in the process. This leads to an interruption in service.

45

A Butterfly network, 10, is shown in Fig. 1 with top ports, 11, connected to external connections, 12, and bottom ports, 13, connected to external connections, 14. A duplicate of this network, 15, with top ports, 16, and bottom ports, 17. A new row, 18, with top ports, 19, and bottom ports, 20, are also shown.

In order to double number of external connections of the Butterfly network, 10, the connections between the top ports, 12, and external connections, 14, have to be broken and connected to the left half of new row bottom ports, 20, and the external connections, 14, connected to the left half of new row top ports, 19. To complete the upgrade, the duplicate network top ports, 16, are connected to the right half of new row bottom ports 17. At this point, duplicate network bottom ports, 17, and right half of new row top ports 19 would be available for new external connections.

The problem is that the connections between the original network top ports, 11, and external connections, 12, have to be disconnected in the process. This leads to an interruption in service.

5 Currently, special "hot slide" multiplexers are designed into switching centers to allow a new switching network to be installed in parallel and then switched in between clock cycles. Unfortunately, this physical electrical switch-over is only part of the problem. The new switch has to have exactly the same "control state" information as the old switch. This requires a considerable amount of hardware and software to accomplish correctly.

100744-02400

Accordingly, the several objects and advantages of my invention are:

- 10

Further objects and advantages of our invention will become apparent from a consideration of the drawings and ensuing description.

Description of Drawings

Fig. 1 shows a 16 port Butterfly multistage switching network being upgraded. (prior art)

Fig. 2 shows a 24 port redundant blocking compensated cyclic group (RBCCG) multistage switching network. (prior art)

Fig. 3A shows a 24 port redundant blocking compensated cyclic group multistage switching network expanded to a 30 port switching network by inserting a new column. The new routers are connected together prior to the insertion.

Fig. 3B shows a 24 port RBCCG multistage switching network expanded to a 30 port switching network by inserting a router after $R(0,3)$ in row 0, after $R(1,3)$ in row 1, between $R(2,2)$ and $R(2,3)$ in row 2, and before $R(3,0)$ in row 3. The new routers are connected together prior to the insertion.

Fig. 3C shows a 24 port RBCCG multistage switching network expanded to a 30 port switching network by inserting a router between $R(0,0)$ and $R(0,1)$ in row 0, after $R(1,3)$ in row 1, between $R(2,2)$ and $R(2,3)$ in row 2, and between $R(3,0)$ and $R(3,1)$ in row 3. Some of the routers are connected together prior to the insertion.

Fig. 4 shows a section of a 30-port RBCCG multistage switching network with the proper interconnection network between two rows of routers.

Fig. 5A, Fig. 5B, Fig. 5C, Fig. 5D, Fig. 5E, Fig. 5F, Fig. 5G, Fig. 5H, Fig. 5I, and Fig. 5J show the connections between router rows $R(1, *)$ and $R(2, *)$ being rewired into a 30 port RBCCG multistage switching network interconnection by moving the connection from bottom port 2 of $R(1,3)$ to bottom port 2 of $R(1,N)$; moving the connection from bottom port 0 of $R(1,3)$ to bottom port 2 of $R(1,3)$; moving the connection from bottom port 1 of $R(1,2)$ to bottom port 0 of $R(1,3)$; moving the connection from bottom port 0 of $R(1,2)$ to bottom port 1 of $R(1,2)$; moving the connection from bottom port 2 of $R(1,1)$ to bottom port 0 of $R(1,2)$; moving the connection from bottom port 1 of $R(1,1)$ to bottom port 2 of $R(1,1)$; moving the connection from bottom port 0 of $R(1,1)$ to bottom port 1 of $R(1,1)$; adding a connection between bottom port 0 of $R(1,1)$ and top port 0 of $R(2,N)$; moving the connection from bottom port 1 of $R(1,3)$ to bottom port 0 of $R(1,N)$; moving the connection from bottom port 2 of $R(1,2)$ to bottom port 1 of $R(1,3)$, respectively.

Fig. 5K shows a partially upgraded 30 port multistage switching network with a completed rewire of the interconnections between router rows $R(1, *)$ and $R(2, *)$ after adding a connection between bottom port 2 of $R(1,2)$ and top port 1 of $R(2,N)$.

Fig. 6A, Fig. 6B, Fig. 6C, Fig. 6D, Fig. 6E, Fig. 6F, Fig. 6G, Fig. 6H, and Fig. 6I show the connections between router rows $R(0, *)$ and $R(1, *)$ being rewired into a 30 port RBCCG multistage switching network interconnection by moving the connection from bottom port 2 of $R(0,3)$ to bottom port 1 of $R(0,N)$; moving the connection from top port 1 of $R(1,0)$ to top port 0 of $R(1,N)$; moving the connection from top port 1 of $R(1,1)$ to top port 1 of $R(1,0)$; moving the connection from top port 1 of $R(1,2)$ to top port 1 of $R(1,1)$; moving the connection from top port 1 of $R(1,3)$ to top port 1 of $R(1,2)$; moving the connection from top port 2 of $R(1,0)$ to top port 1

of $R(1,3)$; moving the connection from top port 2 of $R(1,1)$ to top port 1 of $R(1,N)$; adding a connection between bottom port 2 of $R(0,3)$ and top port 2 of $R(1,1)$; moving the connection from bottom port 1 of $R(0,3)$ to bottom port 0 of $R(0,N)$, respectively.

5 **Fig. 6J** shows a partially upgraded 30 port multistage switching network with a completed rewire of the interconnections between router rows $R(0,*)$ and $R(1,*)$ after adding a connection between bottom port 1 of $R(0,3)$ and top port 2 of $R(1,0)$.

10 **Fig. 7A, Fig. 7B, Fig. 7C, Fig. 7D, Fig. 7E, Fig. 7F, Fig. 7G, Fig. 7H, Fig. 7I, and Fig. 7J** show the connections between router rows $R(2,*)$ and $R(3,*)$ being rewired into a 30 port RBCCG multistage switching network interconnection by moving the connection from top port 0 of $R(3,0)$ to top port 0 of $R(3,N)$; moving the connection from bottom port 1 of $R(2,2)$ to bottom port 0 of $R(2,N)$; moving the connection from top port 1 of $R(3,1)$ to top port 1 of $R(3,N)$; adding a connection between top port 1 of $R(3,1)$ and bottom port 1 of $R(2,2)$; moving the connection from bottom port 2 of $R(2,2)$ to bottom port 2 of $R(2,N)$; moving the connection from top port 0 of $R(3,1)$ to top port 0 of $R(3,0)$; moving the connection from bottom port 0 of $R(2,2)$ to bottom port 2 of $R(2,2)$; moving the connection from top port 0 of $R(3,2)$ to top port 0 of $R(3,1)$; moving the connection from bottom port 1 of $R(2,1)$ to bottom port 0 of $R(2,2)$; moving the connection from top port 0 of $R(3,3)$ to top port 0 of $R(3,2)$, respectively.

Fig. 7K shows a 30 port RBCCG multistage switching after adding a connection between top port 0 of $R(3,3)$ and bottom port 1 of $R(2,1)$.

25 **Fig. 8A, Fig. 8B, Fig. 8C, Fig. 8D, Fig. 8E, Fig. 8F, Fig. 8G, Fig. 8H, Fig. 8I, and Fig. 8J** show the connections between router rows $R(1,*)$ and $R(2,*)$ an being rewired into a 30 port RBCCG multistage switching network interconnection by an alternate method. First, by swapping bottom ports 1 and 2 of $R(0,1)$; swapping bottom ports 0 and 1 of $R(0,2)$; swapping bottom ports 0 and 2 of $R(0,2)$; and finally, swapping bottom ports 0 and 2 of $R(0,3)$, respectively. Second by moving the connection from bottom port 0 of $R(0,3)$ to bottom port 1 of $R(0,N)$; adding a connection between top port 1 of $R(1,N)$ and bottom port 0 of $R(0,3)$; moving the connection from bottom port 1 of $R(0,3)$ to bottom port 0 of $R(0,N)$; moving the connection from top port 1 of $R(1,1)$ to top port 0 of $R(1,N)$; moving the connection from bottom port 0 of $R(0,2)$ to bottom port 1 of $R(0,3)$; adding a connection between bottom port 0 of $R(0,2)$ and top port 1 of $R(1,1)$, respectively.

40 **Fig. 9A, Fig. 9B and Fig. 9C** show the two algorithm for expanding the width of a compensated cyclic group multistage switching network, **Fig. 9A** shows the most general algorithm, while **Fig. 9B** shows a more specific algorithm.

Fig. 10 shows three different algorithms for selecting the insertions positions of new routers into each row. The results are mirrored in **Fig. 3A, Fig. 3B and Fig. 3C**.

45 **Fig. 11A and Fig 11B** show two different algorithms for selecting the order the interconnections should be rewired.

Fig. 12A, Fig. 12B, and Fig. 12C show three different algorithms for selecting which port to rewire during the rewiring phase.

Fig. 13 shows an algorithm for relabeling ports to improve the efficiency of the upgrade process.

Fig. 14 shows the flowchart for the width expansion algorithm shown in **Fig. 9A** and **Fig. 9C**.

Fig. 15 shows the flowchart for the width expansion algorithm shown in **Fig. 9B** and **Fig. 9C**.

Fig. 16A and **Fig. 16B** show the flowcharts for the row selection algorithms shown in **Fig. 11A** and **Fig. 11B** respectively.

Fig. 17 A, Fig. 17B and Fig. 17C show the flowcharts for the port selection algorithms shown in **Fig. 12A, Fig. 12B and Fig 12C**, respectively.

Fig. 18 shows flowchart for the relabeling algorithm shown in **Fig. 13**.

Summary

A multistage switching network can be expanded in width by adding new switching elements to the network without breaking any external connections.

If the switching elements are routers or have the ability to route traffic the expansion may be performed without severely disrupting its ability to route traffic provided that the network has adequate path redundancy.

The width augmentation process has three constituent components. The first component is the selection of the insertion points of the new routers into each row. The positions can be arbitrary, but it is best to insert them at the end of each row, since this minimizes the need to use the path redundancy to route traffic around an upgrade induced fault.

The second component has two types of steps of which one is optional. The first type is to select a port that is misconnected in reference to the desired topology and to rewire it to its appropriate counterpart. The second type is optional and is the permutation of ports on the same router which results in a correct port being connected to the correct router. If such a permutation can be performed logically, traffic need not be disrupted. These two types of steps are repeated in any order until the desired topology is achieved. Since the ordering of such steps can be arbitrary, more structure can be added by breaking this component into three subcomponents.

The first subcomponent is the selection of the order in which each interconnection network is rewired into the desired topology. The order can be arbitrary, however, since the path redundancy is greatest in the middle of the network, the best order has been found to start with the middle row or the closest to the middle. The subsequent rows should be ordered with those closest to the middle first and those farthest away from the middle last.

The second subcomponent, which is optional, is the permutation of ports described above. This step should always be taken before rewiring an interconnection network, if such a permutation can be performed logically so that no traffic disruption occurs. This step minimizes bandwidth reduction during the rewiring of an interconnection network.

The third subcomponent is the rewiring phase. This is accomplished by systematically disconnecting from a selected port any misconnected connection to that port and to the corresponding port to which the selected port is to be connected. With any misconnections disconnected, the selected port and its appropriate corresponding port may then be connected. In order to minimize the impact on the throughput bandwidth, priority in the port selection process should be given to selecting ports not currently connected. This introduces a plethora of port selection algorithms. Additional preferences, contribute to better throughput bandwidth during the upgrade process.

The final component is relatively simple. All external connections that are not already connected should be connected and placed into service.

Description of Invention

A redundant blocking compensated cyclic group (RBCCG) multistage network, 30, is shown in Fig. 2 and discussed further in United States Patent 5,841,775, "Scalable Switching Networks" by Alan Huang, Nov. 24, 1998. It consists of rows, 31, 33, 35, and 37 of routers or other switching elements. These rows of routers are connected together via interconnection networks 32, 34, and 36. The routers are designed $R(\text{row}, \text{column})$ where the top left most router is denoted $R(0,0)$. The top ports of each router are numbered from left to right starting with 0. The bottom ports of each router are numbered from left to right starting with 0. The top ports of each interconnection network are numbered from left to right starting with 0. The bottom ports of each interconnection network are numbered from left to right starting with 0.

The width of the RBCCG can be expanded by inserting a new router $R(0,N)$ into row 31 of routers in any position; inserting a new router $R(1,N)$ into row 33 of routers in any position; inserting a new router $R(2,N)$ into row 35 of routers in any position; inserting a new router $R(3,N)$ into row 37 of routers in any position. The upgrade can be accomplished by rewiring interconnection networks 32, 34, and 36, without breaking the connections between the top ports of router row 31 and external connections 38 or breaking the connections between the bottom ports of router row 37 and external connections 39. Three examples of insertion positions are shown in Fig. 3A, Fig. 3B, and Fig. 3C.

The most general form of the upgrade process is given by Fig. 9A, with supporting functions given in Fig. 9C, and flowcharted in Fig. 14. The first major step is to insert the desired new routers into each row. Though the algorithm depicted goes row by row, the order may be arbitrary. Since the ordering does not impact quality factors in the upgrade process, a row by row order is used.

The particular position of the new routers in each row can be arbitrary. The "insertion_position" algorithm consists of: Selecting a position in each row to insert a new router. For each row called *current_row*, if a connection be made that matches the interconnection pattern of the upgraded network between $R(\text{current_row}, N)$ and $R(\text{current_row}+1, N)$ connect the appropriate bottom port of $R(\text{current_row}, N)$ and the appropriate top port of $R(\text{current_row}+1, N)$. If no connection can be made, then the bottom ports of $R(\text{current_row}, N)$ and the top ports of $R(\text{current_row}+1, N)$ are left alone. A general "insertion_position" algorithm is shown in Fig. 10, along with two specific "insertion_position" algorithms that are used in the examples to come.

The upgrade process continues after the routers are added by selecting any port which is misconnected with respect to the desired final topology. The order of selection does have an impact on the disruption of traffic and even the number of total steps. A more systematic and organized approach is given below in a refinement of this algorithm. Once a port is selected it is rewired by connecting it to its appropriate port with respect to the final topology. Henceforth, this port will be known as its corresponding port. The remainder of the algorithm in Fig. 9A and flowchart Fig. 14 is a detailed description of the rewiring based on the connection states of the port and its corresponding port.

There is also an optional branch in the upgrade process. In scanning the routers, one may discover routers for which some connections originate from the correct routers (with respect to the desired final topology) but connect to the wrong port. If a logical reassignment of port

addresses were possible, achieving a virtual port permutation while leaving the physical connections unchanged. This would reduce the number of connections that need to be broken. This step as indicated by the optional branch may be executed at any time, that is it need not be executed in every loop of the algorithm. For instance, the sequence could be rewire, relabel, relabel, relabel, rewire, rewire, relabel, rewire, etc. A more systematic approach is given in a refined version of the upgrade algorithm.

The final step in this process is to connect and activate any new external ports that result from the addition of new ports in this upgrade process.

The upgrade process is refined in Fig. 9B with flowchart in Fig. 15. The first part with depicting the router insertion and the last step involving the activation of new external ports are the same as the more general algorithm. The distinction is that the rewiring and the optional relabeling is made more systematic. The first refinement is that the process takes place row by row or more precisely interconnection network by interconnection network. This is illustrated by the creation of the "row_select" component. Once a row is selected the optional "relabel_ports" component may be executed for that row. After that, a port that is part of the selected interconnection network, by convention the interconnection network below the selected row, is selected by the "select_port" component and rewired to its corresponding port. These three component algorithms are given in more detail below.

In the examples given below, the process of "diverting traffic" away from a port and "stop diverting traffic" from a port (also called "allow traffic to flow" to a port as stated in Fig. 9C.) should occur automatically whenever a connection is broken (disconnected) and when a connection is established, respectively if the routers are running standard routing protocols such as OSPF. However, the disconnection process may be streamlined by forcing new routing table information to be the routers prior to the connection being broken, thereby speeding the upgrade process and minimizing the impact on traffic. Likewise, a forced change to the routing table may expedite the full use of bandwidth after a new connection is established. In addition, to streamline the notation in the flowcharts, Fig. 14 and Fig. 15, the act of disconnection also implies the diversion of traffic from its attached ports prior to the breaking of the connection. Also, the act of connecting two ports implies the allowing of new traffic flow through the two ports (or equivalently the "stopping of the diversion of traffic" from the two ports) after the attachment of a connection between the two ports.

Various examples of the upgrade process are given in greater detail. Fig. 3A shows insertion of the new routers as a column with the new routers added to the end of each row. Fig. 3B and Fig. 3C show the width of each row extended by inserting new routers in various positions within the existing row. Fig. 4 shows the generic interconnection network of a 30-port RBCCG multistage network. This serves as an interconnection template where 61 represents at any given time router row, current_row, and 63, router row, current_row+1, with 62 as the correct interconnection network for a 30-port RBCCG multistage network. In the process of upgrading, interconnection networks 42, 43, and 44 of Fig. 3A will each have to be rewired to look like 62 of Fig. 4. Similarly, interconnection networks 47, 48, and 49 of Fig. 3B and interconnection networks 52, 53 and 54 will each have to be rewired to look like 62 of Fig. 4 for their respective subsequent upgrade steps.

Fig. 3A shows a connection between bottom port 2 of $R(0,N)$ and top port 2 of $R(1,N)$, a connection between bottom port 2 of $R(1,N)$ and top port 2 of $R(2,N)$ and a connection between bottom port 2 of $R(2,N)$ and top port 2 of $R(3,N)$, because their respective connections are the same as those in 62 of **Fig. 4**.

Fig. 3B shows a connection between bottom port 2 of $R(0,N)$ and top port 2 of $R(1,N)$, a connection between bottom port 1 of $R(1,N)$ and top port 2 of $R(2,N)$ and a connection between bottom port 1 of $R(2,N)$ and top port 2 of $R(3,N)$, because their respective connections are the same as those in 62 of **Fig. 4**.

Fig. 3C shows no connection between $R(0,N)$ and $R(1,N)$, a connection between bottom port 1 of $R(1,N)$ and top port 2 of $R(2,N)$ and a connection between bottom port 2 of $R(2,N)$ and top port 2 of $R(3,N)$, because their respective connections are the same as those in 62 of **Fig. 4**. The lack of connection between $R(0,N)$ and $R(1,N)$ is due to the lack of a corresponding connection in 62 of **Fig. 4**.

While all insertion patterns will yield a viable width upgrade, the impact on disruption of throughput bandwidth can vary. Of the three examples illustrated, **Fig. 3C** is the least desirable since there is not a connection that can be made between $R(0,N)$ and $R(1,N)$ that would not need to be rewired later in the rewiring phase. The insertion illustrated in **Fig. 3A** and **Fig. 3B** are more desirable because a connection between $R(n,N)$ and $R(n+1,N)$ can be made prior to the rewiring phase. Of the two, **Fig. 3A** has the least impact on the throughput bandwidth. However, the subsequent description of this algorithm will focus on **Fig. 3B** because it illustrates the more general upgrade methodology more diversely.

Once the new routers are inserted. The examples follow the algorithm given in **Fig. 9B** with examples of the “row_select” and “select_port” components defined and illustrated below.

The “row_select” algorithm selects a sequencing of the interconnections between rows. Each interconnection shall be called by the row number, whose bottom ports the interconnections are attached to. This sequence must include all the interconnections, but may be in any order. **Fig. 11A** shows the best “row_select” algorithm. **Fig. 11B** shows another common “row_select” algorithm. **Fig. 16A** and **Fig. 16B** show the respective flowcharts.

There is an optional relabel_ports stage, which will be described later. The remaining component of the “upgrade” algorithm is the “select_port” port algorithm. During the rewiring of each interconnection network, the basic “upgrade” algorithm calls for the selection of any port, top or bottom not properly connected to it’s appropriate corresponding port in accordance with 62 of **Fig. 4**. Though any port may be selected at each step, it has been found that priority should be given to ports that are not already connected. If a port that is already connected is selected, it may introduce an additional disconnection resulting in degradation of throughput bandwidth. **Fig. 12A**, **Fig. 12B** and **Fig. 12C** show three common “select_port” algorithms. Corresponding flowcharts are shown in **Fig. 17A**, **Fig. 17B**, and **Fig. 17C**.

The upgrade using **Fig. 3B** as an example is shown in much greater detail. For the sake of diversity of the algorithm, three different port selection algorithms are used for each row. The “row_select” algorithm shown in **Fig. 11A** is used.

In accordance with this “row_select” algorithm, the first row to be selected as the row in our example is row 1, that is interconnection network, 48, in Fig. 3B.

In the rewiring of interconnection network, 48, the “select_port” algorithm shown in Fig. 12A is employed. The process begins by selecting bottom port 2 of R(1,N) because scanning from right to left it is the first bottom port not connected to its corresponding port, top port 2 of R(2,3), according to Fig. 4. Top port 2 of R(2,3) is currently connected to bottom port 2 of R(1,3). Divert traffic from top port 2 of R(2,3) and bottom port 2 of R(1,3). Stop top port 2 of R(2,3) and bottom port 2 of R(1,3). Disconnect top port 2 of R(2,3) and bottom port 2 of R(1,3) and move the disconnected connection to bottom port 2 of R(1,N) as shown in Fig. 5A. Start bottom port 2 of R(1,N) and top port 2 of R(2,3). Stop diverting the traffic from bottom port 2 of R(1,N) and top port 2 of R(2,3).

The process continues by selecting bottom port 2 of R(1,3) because it was disconnected in the previous step shown in Fig. 5A. Bottom port 2 of R(1,3) should be connected to top port 2 of R(2,1) according to Fig. 4. Top port 2 of R(2,1) is currently connected to bottom port 0 of R(1,3). Divert traffic from top port 2 of R(2,1) and bottom port 0 of R(1,3). Stop top port 2 of R(2,1) and bottom port 0 of R(1,3). Disconnect top port 2 of R(2,1) and bottom port 0 of R(1,3) and move the disconnected connection to bottom port 2 of R(1,3) as shown in Fig. 5B. Start bottom port 2 of R(1,3) and top port 2 of R(2,1). Stop diverting the traffic from bottom port 2 of R(1,3) and top port 2 of R(2,1).

The process continues by selecting bottom port 0 of R(1,3) because it was disconnected in the previous step shown in Fig. 5B. Bottom port 0 of R(1,3) should be connected to top port 1 of R(2,3) according to Fig. 4. Top port 1 of R(2,3) is currently connected to bottom port 1 of R(1,2). Divert traffic from top port 1 of R(2,3) and bottom port 1 of R(1,2). Stop top port 1 of R(2,3) and bottom port 1 of R(1,2). Disconnect top port 1 of R(2,3) and bottom port 1 of R(1,2) and move the disconnected connection to bottom port 0 of R(1,3) as shown in Fig. 5C. Start bottom port 0 of R(1,3) and top port 1 of R(2,3). Stop diverting the traffic from bottom port 0 of R(1,3) and top port 1 of R(2,3).

The process continues by selecting bottom port 1 of R(1,2) because it was disconnected in the previous step shown in Fig. 5C. Bottom port 1 of R(1,2) should be connected to top port 1 of R(2,2) according to Fig. 4. Top port 1 of R(2,2) is currently connected to bottom port 0 of R(1,2). Divert traffic from top port 1 of R(2,2) and bottom port 0 of R(1,2). Stop top port 1 of R(2,2) and bottom port 0 of R(1,2). Disconnect top port 1 of R(2,2) and bottom port 0 of R(1,2) and move the disconnected connection to bottom port 1 of R(1,2) as shown in Fig. 5D. Start bottom port 1 of R(1,2) and top port 1 of R(2,2). Stop diverting the traffic from bottom port 1 of R(1,2) and top port 1 of R(2,2).

The process continues by selecting bottom port 0 of R(1,2) because it was disconnected in the previous step shown in Fig. 5D. Bottom port 0 of R(1,2) should be connected to top port 1 of R(2,1) according to Fig. 4. Top port 1 of R(2,1) is currently connected to bottom port 2 of R(1,1). Divert traffic from top port 1 of R(2,1) and bottom port 2 of R(1,1). Stop top port 1 of R(2,1) and bottom port 2 of R(1,1). Disconnect top port 1 of R(2,1) and bottom port 2 of R(1,1) and move the disconnected connection to bottom port 0 of R(1,2) as shown in Fig. 5E. Start bottom port 0 of R(1,2) and top port 1 of R(2,1). Stop diverting the traffic from bottom port 0 of R(1,2) and top port 1 of R(2,1).

The process continues by selecting bottom port 2 of R(1,1) because it was disconnected in the previous step shown in Fig. 5E. Bottom port 2 of R(1,1) should be connected to top port 1 of R(2,0) according to Fig. 4. Top port 1 of R(2,0) is currently connected to bottom port 1 of R(1,1). Divert traffic from top port 1 of R(2,0) and bottom port 1 of R(1,1). Stop top port 1 of R(2,0) and bottom port 1 of R(1,1). Disconnect top port 1 of R(2,0) and bottom port 1 of R(1,1) and move the disconnected connection to bottom port 2 of R(1,1) as shown in Fig. 5F. Start bottom port 2 of R(1,1) and top port 1 of R(2,0). Stop diverting the traffic from bottom port 2 of R(1,1) and top port 1 of R(2,0).

The process continues by selecting bottom port 1 of R(1,1) because it was disconnected in the previous step shown in Fig. 5F. Bottom port 1 of R(1,1) should be connected to top port 0 of R(2,3) according to Fig. 4. Top port 0 of R(2,3) is currently connected to bottom port 0 of R(1,1). Divert traffic from top port 0 of R(2,3) and bottom port 0 of R(1,1). Stop top port 0 of R(2,3) and bottom port 0 of R(1,1). Disconnect top port 0 of R(2,3) and bottom port 0 of R(1,1) and move the disconnected connection to bottom port 1 of R(1,1) as shown in Fig. 5G. Start bottom port 1 of R(1,1) and top port 0 of R(2,3). Stop diverting the traffic from bottom port 1 of R(1,1) and top port 0 of R(2,3).

The process continues by selecting bottom port 0 of R(1,1) because it was disconnected in the previous step shown in Fig. 5G. Bottom port 0 of R(1,1) should be connected to top port 0 of R(2,N) according to Fig. 4. Top port 0 of R(2,N) is currently not connected. Establish a new connection between bottom port 0 of R(1,1) and top port 0 of R(2,N) as shown in Fig. 5H. Start bottom port 0 of R(1,1) and top port 0 of R(2,N). Stop diverting the traffic from bottom port 0 of R(1,1) and top port 0 of R(2,N).

The process continues by selecting bottom port 0 of R(1,N) because scanning from right to left it is the first bottom port not connected to it's corresponding port, top port 2 of R(2,2), according to Fig. 4. Top port 2 of R(2,2) is currently connected to bottom port 1 of R(1,3). Divert traffic from top port 2 of R(2,2) and bottom port 1 of R(1,3). Stop top port 2 of R(2,2) and bottom port 1 of R(1,3). Disconnect top port 2 of R(2,2) and bottom port 1 of R(1,3) and move the disconnected connection to bottom port 0 of R(1,N) as shown in Fig. 5I. Start bottom port 0 of R(1,N) and top port 2 of R(2,2). Stop diverting the traffic from bottom port 0 of R(1,N) and top port 2 of R(2,2).

The process continues by selecting bottom port 1 of R(1,3) because it was disconnected in the previous step shown in Fig. 5I. Bottom port 1 of R(1,3) should be connected to top port 2 of R(2,0) according to Fig. 4. Top port 2 of R(2,0) is currently connected to bottom port 2 of R(1,2). Divert traffic from top port 2 of R(2,0) and bottom port 2 of R(1,2). Stop top port 2 of R(2,0) and bottom port 2 of R(1,2). Disconnect top port 2 of R(2,0) and bottom port 2 of R(1,2) and move the disconnected connection to bottom port 1 of R(1,3) as shown in Fig. 5J. Start bottom port 1 of R(1,3) and top port 2 of R(2,0). Stop diverting the traffic from bottom port 1 of R(1,3) and top port 2 of R(2,0).

The process continues by selecting bottom port 2 of R(1,2) because it was disconnected in the previous step shown in Fig. 5J. Bottom port 2 of R(1,2) should be connected to top port 1 of R(2,N) according to Fig. 4. Top port 1 of R(2,N) is currently not connected. Establish a new connection between bottom port 2 of R(1,2) and top port 1 of R(2,N) as shown in Fig. 5K. Start bottom port 2 of R(1,2) and top port 1 of R(2,N). Stop diverting the traffic from bottom port 2 of R(1,2) and top port 1 of R(2,N).

R(1,2) and top port 1 of R(2,N). This completes the rewiring of the interconnection network, 48, between row 1 and row 2.

In accordance with the "row_select" algorithm of Fig. 11A, the second row to be selected as the row in our example is row 0, that is interconnection network, 47 in Fig. 3B.

In the rewiring of interconnection network, 47, the "select_port" algorithm shown in Fig. 12B is employed. This algorithm is substantially more complex than that in used previously in Fig. 12A, so at each step, the selection process is described in much detail first before the rewiring is described in detail.

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, bottom port 1 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,3). Top port 2 of R(1,3) is currently connected to bottom port 2 of R(0,3). The breaking of this connection does not leave router R(0,3) or router R(1,3) with more than one broken connection. This completes the selection process for this step.

With bottom port 1 of R(0,N) selected and recalling that its corresponding port, top port 2 of R(1,3), is connected to bottom port 2 of R(0,3), the process continues by diverting traffic from top port 2 of R(1,3) and bottom port 2 of R(0,3); stopping top port 2 of R(1,3) and bottom port 2 of R(0,3); disconnecting top port 2 of R(1,3) and bottom port 2 of R(0,3) and moving the disconnected connection to bottom port 1 of R(0,N) as shown in Fig. 6A; starting bottom port 1 of R(0,N) and top port 2 of R(1,3); and stop diverting the traffic from bottom port 1 of R(0,N) and top port 2 of R(1,3).

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of R(1,N) is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of R(0,3). Bottom port 0 of R(0,3) is currently connected to top port 2 of R(1,1). The breaking this connection introduces a second broken connection to R(0,3), so top port 1 of R(1,N) is not selected. Continuing to scan from right to left, bottom port 0 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,2). Top port 2 of R(1,2) is currently connected to bottom port 1 of R(0,3). The breaking this connection introduces a second broken connection to R(0,3), so bottom port 0 of R(0,N) is not selected. Continuing to scan from right to left, top port 0 of R(1,N) is not connected to anything with corresponding port according to Fig. 4, bottom port 1 of R(0,1). Bottom port 1 of R(0,1) is currently connected to top port 1 of R(1,0). The breaking of this connection does not leave router R(0,1) or router R(1,0) with more than one broken connection. This completes the selection process for this step.

With top port 0 of R(1,N) selected and recalling that its corresponding port, bottom port 1 of R(0,1), is connected to top port 1 of R(1,0), the process continues by diverting traffic from bottom port 1 of R(0,1) and top port 1 of R(1,0); stopping bottom port 1 of R(0,1) and top port 1 of R(1,0); disconnecting bottom port 1 of R(0,1) and top port 1 of R(1,0) and moving the disconnected connection to top port 0 of R(1,N) as shown in Fig. 6B; starting top port 0 of R(1,N) and bottom port 1 of R(0,1); and stop diverting the traffic from top port 0 of R(1,N) and bottom port 1 of R(0,1).

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of $R(1,N)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of $R(0,3)$. Bottom port 0 of $R(0,3)$ is currently connected to top port 2 of $R(1,1)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so top port 1 of $R(1,N)$ is not selected. Continuing to scan from right to left, bottom port 0 of $R(0,N)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,2)$. Top port 2 of $R(1,2)$ is currently connected to bottom port 1 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 0 of $R(0,N)$ is not selected. Continuing to scan from right to left, bottom port 2 of $R(0,3)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,1)$. Top port 2 of $R(1,1)$ is currently connected to bottom port 0 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 2 of $R(0,3)$ is not selected. Continuing to scan from right to left, top port 1 of $R(1,0)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 2 of $R(0,1)$. Bottom port 2 of $R(0,1)$ is currently connected to top port 1 of $R(1,1)$. The breaking of this connection does not leave router $R(0,1)$ or router $R(1,1)$ with more than one broken connection. This completes the selection process for this step.

With top port 1 of $R(1,0)$ selected and recalling that its corresponding port, bottom port 2 of $R(0,1)$, is connected to top port 1 of $R(1,1)$, the process continues by diverting traffic from bottom port 2 of $R(0,1)$ and top port 1 of $R(1,1)$; stopping bottom port 2 of $R(0,1)$ and top port 1 of $R(1,1)$; disconnecting bottom port 2 of $R(0,1)$ and top port 1 of $R(1,1)$ and moving the disconnected connection to top port 1 of $R(1,0)$ as shown in Fig. 6C; starting top port 1 of $R(1,0)$ and bottom port 2 of $R(0,1)$; and stop diverting the traffic from top port 1 of $R(1,0)$ and bottom port 2 of $R(0,1)$.

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of $R(1,N)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of $R(0,3)$. Bottom port 0 of $R(0,3)$ is currently connected to top port 2 of $R(1,1)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so top port 1 of $R(1,N)$ is not selected. Continuing to scan from right to left, bottom port 0 of $R(0,N)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,2)$. Top port 2 of $R(1,2)$ is currently connected to bottom port 1 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 0 of $R(0,N)$ is not selected. Continuing to scan from right to left, bottom port 2 of $R(0,3)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,1)$. Top port 2 of $R(1,1)$ is currently connected to bottom port 0 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 2 of $R(0,3)$ is not selected. Continuing to scan from right to left, top port 1 of $R(1,1)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of $R(0,2)$. Bottom port 0 of $R(0,2)$ is currently connected to top port 1 of $R(1,2)$. The breaking of this connection does not leave router $R(0,2)$ or router $R(1,2)$ with more than one broken connection. This completes the selection process for this step.

With top port 1 of $R(1,1)$ selected and recalling that its corresponding port, bottom port 0 of $R(0,2)$, is connected to top port 1 of $R(1,2)$, the process continues by diverting traffic from bottom port 0 of $R(0,2)$ and top port 1 of $R(1,2)$; stopping bottom port 0 of $R(0,2)$ and top port 1 of $R(1,2)$; disconnecting bottom port 0 of $R(0,2)$ and top port 1 of $R(1,2)$ and moving the

disconnected connection to top port 1 of R(1,1) as shown in Fig. 6D; starting top port 1 of R(1,1) and bottom port 0 of R(0,2); and stop diverting the traffic from top port 1 of R(1,1) and bottom port 0 of R(0,2).

5 Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of R(1,N) is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of R(0,3). Bottom port 0 of R(0,3) is currently connected to top port 2 of R(1,1). The breaking this connection introduces a second broken connection to R(0,3), so top port 1 of R(1,N) is not selected. Continuing to scan from right to left, bottom port 0 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,2). Top port 2 of R(1,2) is currently connected to bottom port 1 of R(0,3). The breaking this connection introduces a second broken connection to R(0,3), so bottom port 0 of R(0,N) is not selected. Continuing to scan from right to left, bottom port 2 of R(0,3) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,1). Top port 2 of R(1,1) is currently connected to bottom port 0 of R(0,3). The breaking this connection introduces a second broken connection to R(0,3), so bottom port 2 of R(0,3) is not selected. Continuing to scan from right to left, top port 1 of R(1,2) is not connected to anything with corresponding port according to Fig. 4, bottom port 1 of R(0,2). Bottom port 1 of R(0,2) is currently connected to top port 1 of R(1,3). The breaking of this connection does not leave router R(0,2) or router R(1,3) with more than one broken connection. This completes the selection process for this step.

With top port 1 of R(1,2) selected and recalling that its corresponding port, bottom port 1 of R(0,2), is connected to top port 1 of R(1,3), the process continues by diverting traffic from bottom port 1 of R(0,2) and top port 1 of R(1,3); stopping bottom port 1 of R(0,2) and top port 1 of R(1,3); disconnecting bottom port 1 of R(0,2) and top port 1 of R(1,3) and moving the disconnected connection to top port 1 of R(1,2) as shown in Fig. 6E; starting top port 1 of R(1,2) and bottom port 1 of R(0,2); and stop diverting the traffic from top port 1 of R(1,2) and bottom port 1 of R(0,2).

30 Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of R(1,N) is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of R(0,3). Bottom port 0 of R(0,3) is currently connected to top port 2 of R(1,1). The breaking this connection introduces a second broken connection to R(0,3), so top port 1 of R(1,N) is not selected. Continuing to scan from right to left, bottom port 0 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,2). Top port 2 of R(1,2) is currently connected to bottom port 1 of R(0,3). The breaking this connection introduces a second broken connection to R(0,3), so bottom port 0 of R(0,N) is not selected. Continuing to scan from right to left, bottom port 2 of R(0,3) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,1). Top port 2 of R(1,1) is currently connected to bottom port 0 of R(0,3). The breaking this connection introduces a second broken connection to R(0,3), so bottom port 2 of R(0,3) is not selected. Continuing to scan from right to left, top port 1 of R(1,3) is not connected to anything with corresponding port according to Fig. 4, bottom port 2 of R(0,2). Bottom port 2 of R(0,2) is currently connected to top port 2 of R(1,0). The breaking of this connection does not leave router R(0,2) or router R(1,0) with more than one broken connection. This completes the selection process for this step.

With top port 1 of $R(1,3)$ selected and recalling that its corresponding port, bottom port 2 of $R(0,2)$, is connected to top port 2 of $R(1,0)$, the process continues by diverting traffic from bottom port 2 of $R(0,2)$ and top port 2 of $R(1,0)$; stopping bottom port 2 of $R(0,2)$ and top port 2 of $R(1,0)$; disconnecting bottom port 2 of $R(0,2)$ and top port 2 of $R(1,0)$ and moving the disconnected connection to top port 1 of $R(1,3)$ as shown in Fig. 6F; starting top port 1 of $R(1,3)$ and bottom port 2 of $R(0,2)$; and stop diverting the traffic from top port 1 of $R(1,3)$ and bottom port 2 of $R(0,2)$.

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 1 of $R(1,N)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 0 of $R(0,3)$. Bottom port 0 of $R(0,3)$ is currently connected to top port 2 of $R(1,1)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so top port 1 of $R(1,N)$ is not selected. Continuing to scan from right to left, bottom port 0 of $R(0,N)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,2)$. Top port 2 of $R(1,2)$ is currently connected to bottom port 1 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 0 of $R(0,N)$ is not selected. Continuing to scan from right to left, bottom port 2 of $R(0,3)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,1)$. Top port 2 of $R(1,1)$ is currently connected to bottom port 0 of $R(0,3)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so bottom port 2 of $R(0,3)$ is not selected. Continuing to scan from right to left, top port 2 of $R(1,0)$ is not connected to anything with corresponding port according to Fig. 4, bottom port 1 of $R(0,3)$. Bottom port 1 of $R(0,3)$ is currently connected to top port 2 of $R(1,2)$. The breaking this connection introduces a second broken connection to $R(0,3)$, so top port 2 of $R(1,0)$ is not selected. Since none of the ports that are not connected satisfy the second port selection criterion, the selection process continues by selecting the rightmost port not connected to anything, that is top port 1 of $R(1,N)$.

With top port 1 of $R(1,N)$ selected and recalling that its corresponding port, bottom port 0 of $R(0,3)$, is connected to top port 2 of $R(1,1)$, the process continues by diverting traffic from bottom port 0 of $R(0,3)$ and top port 2 of $R(1,1)$; stopping bottom port 0 of $R(0,3)$ and top port 2 of $R(1,1)$; disconnecting bottom port 0 of $R(0,3)$ and top port 2 of $R(1,1)$ and moving the disconnected connection to top port 1 of $R(1,N)$ as shown in Fig. 6G; starting top port 1 of $R(1,N)$ and bottom port 0 of $R(0,3)$; and stop diverting the traffic from top port 1 of $R(1,N)$ and bottom port 0 of $R(0,3)$.

The process continues by selecting bottom port 2 of $R(0,3)$ because scanning from right to left, it is the first port bottom or top not connected to anything and whose corresponding port (according to Fig. 4) is also not connected to anything; establishing a new connection between bottom port 2 of $R(0,3)$ and top port 2 of $R(1,1)$ as shown in Fig. 6H; starting bottom port 2 of $R(0,3)$ and top port 2 of $R(1,1)$; and stop diverting the traffic from bottom port 2 of $R(0,3)$ and top port 2 of $R(1,1)$.

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, bottom port 0 of $R(0,N)$ is not connected to anything with corresponding port according to Fig. 4, top port 2 of $R(1,2)$. Top port 2 of $R(1,2)$ is currently connected to bottom port 1 of $R(0,3)$. The breaking of this connection does not leave router $R(0,3)$ or router $R(1,2)$ with more than one broken connection. This completes the selection process for this step.

With bottom port 0 of $R(0,N)$ selected and recalling that its corresponding port, top port 2 of $R(1,2)$, is connected to bottom port 1 of $R(0,3)$, the process continues by diverting traffic from top port 2 of $R(1,2)$ and bottom port 1 of $R(0,3)$; stopping top port 2 of $R(1,2)$ and bottom port 1 of $R(0,3)$; disconnecting top port 2 of $R(1,2)$ and bottom port 1 of $R(0,3)$ and moving the disconnected connection to bottom port 0 of $R(0,N)$ as shown in Fig. 6I; starting bottom port 0 of $R(0,N)$ and top port 2 of $R(1,2)$; and stop diverting the traffic from bottom port 0 of $R(0,N)$ and top port 2 of $R(1,2)$.

The process continues by selecting bottom port 1 of $R(0,3)$ because scanning from right to left, it is the first port bottom or top not connected to anything and whose corresponding port (according to Fig. 4) is also not connected to anything; establishing a new connection between bottom port 1 of $R(0,3)$ and top port 2 of $R(1,0)$ as shown in Fig. 6J; starting bottom port 1 of $R(0,3)$ and top port 2 of $R(1,0)$; and stop diverting the traffic from bottom port 1 of $R(0,3)$ and top port 2 of $R(1,0)$. This completes the rewiring of interconnection network, 47, between row 0 and row 1.

In accordance with the "row_select" algorithm of Fig. 11A, the third row to be selected as the row in our example is row 2, that is interconnection network, 49, in Fig. 3B.

In the rewiring of interconnection network, 49, the "select_port" algorithm shown in Fig. 12C is employed. This algorithm is intended to show a more arbitrary method of selecting ports that still yields good results. In addition when loading up the "port fifo" we alternate scanning left to right for top ports then bottom ports that aren't.

The process begins by loading the port fifo with {top port 0 of $R(3,N)$, bottom port 0 of $R(2,N)$, top port 1 of $R(3,N)$, bottom port 2 of $R(2,N)$ }.

The process continues by selecting top port 0 of $R(3,N)$ because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Top port 0 of $R(3,N)$ should be connected to bottom port 0 of $R(2,0)$ according to Fig. 4. Bottom port 0 of $R(2,0)$ is currently connected to top port 0 of $R(3,0)$. Divert traffic from bottom port 0 of $R(2,0)$ and top port 0 of $R(3,0)$. Stop bottom port 0 of $R(2,0)$ and top port 0 of $R(3,0)$. Disconnect bottom port 0 of $R(2,0)$ and top port 0 of $R(3,0)$ and move the disconnected connection to top port 0 of $R(3,N)$ as shown in Fig. 7A. Start top port 0 of $R(3,N)$ and bottom port 0 of $R(2,0)$. Stop diverting the traffic from top port 0 of $R(3,N)$ and bottom port 0 of $R(2,0)$.

The port fifo now contains {bottom port 0 of $R(2,N)$, top port 1 of $R(3,N)$, bottom port 2 of $R(2,N)$ }. The process continues by selecting bottom port 0 of $R(2,N)$ because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Bottom port 0 of $R(2,N)$ should be connected to top port 1 of $R(3,3)$ according to Fig. 4. Top port 1 of $R(3,3)$ is currently connected to bottom port 1 of $R(2,2)$. Divert traffic from top port 1 of $R(3,3)$ and bottom port 1 of $R(2,2)$. Stop top port 1 of $R(3,3)$ and bottom port 1 of $R(2,2)$. Disconnect top port 1 of $R(3,3)$ and bottom port 1 of $R(2,2)$ and move the disconnected connection to bottom port 0 of $R(2,N)$ as shown in Fig. 7B. Start bottom port 0 of $R(2,N)$ and top port 1 of $R(3,3)$. Stop diverting the traffic from bottom port 0 of $R(2,N)$ and top port 1 of $R(3,3)$.

The port fifo now contains {top port 1 of R(3,N), bottom port 2 of R(2,N)}. The process continues by selecting top port 1 of R(3,N) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Top port 1 of R(3,N) should be connected to bottom port 2 of R(2,1) according to Fig. 4. Bottom port 2 of R(2,1) is currently connected to top port 1 of R(3,1). Divert traffic from bottom port 2 of R(2,1) and top port 1 of R(3,1). Stop bottom port 2 of R(2,1) and top port 1 of R(3,1). Disconnect bottom port 2 of R(2,1) and top port 1 of R(3,1) and move the disconnected connection to top port 1 of R(3,N) as shown in Fig. 7C. Start top port 1 of R(3,N) and bottom port 2 of R(2,1). Stop diverting the traffic from top port 1 of R(3,N) and bottom port 2 of R(2,1).

The process continues by selecting top port 1 of R(3,1) because scanning from right to left, it is the first port bottom or top not connected to anything and whose corresponding port (according to Fig. 4) is also not connected to anything. Establish a new connection between top port 1 of R(3,1) and bottom port 1 of R(2,2) as shown in Fig. 7D. Start top port 1 of R(3,1) and bottom port 1 of R(2,2). Stop diverting the traffic from top port 1 of R(3,1) and bottom port 1 of R(2,2).

The port fifo now contains {bottom port 2 of R(2,N)}. The process continues by selecting bottom port 2 of R(2,N) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Bottom port 2 of R(2,N) should be connected to top port 2 of R(3,0) according to Fig. 4. Top port 2 of R(3,0) is currently connected to bottom port 2 of R(2,2). Divert traffic from top port 2 of R(3,0) and bottom port 2 of R(2,2). Stop top port 2 of R(3,0) and bottom port 2 of R(2,2). Disconnect top port 2 of R(3,0) and bottom port 2 of R(2,2) and move the disconnected connection to bottom port 2 of R(2,N) as shown in Fig. 7E. Start bottom port 2 of R(2,N) and top port 2 of R(3,0). Stop diverting the traffic from bottom port 2 of R(2,N) and top port 2 of R(3,0).

The port fifo is now empty. The process continues by loading the port fifo with {top port 0 of R(3,0), bottom port 2 of R(2,2)}.

The process continues by selecting top port 0 of R(3,0) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Top port 0 of R(3,0) should be connected to bottom port 1 of R(2,0) according to Fig. 4. Bottom port 1 of R(2,0) is currently connected to top port 0 of R(3,1). Divert traffic from bottom port 1 of R(2,0) and top port 0 of R(3,1). Stop bottom port 1 of R(2,0) and top port 0 of R(3,1). Disconnect bottom port 1 of R(2,0) and top port 0 of R(3,1) and move the disconnected connection to top port 0 of R(3,0) as shown in Fig. 7F. Start top port 0 of R(3,0) and bottom port 1 of R(2,0). Stop diverting the traffic from top port 0 of R(3,0) and bottom port 1 of R(2,0).

The port fifo now contains {bottom port 2 of R(2,2)}. The process continues by selecting bottom port 2 of R(2,2) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Bottom port 2 of R(2,2) should be connected to top port 1 of R(3,2) according to Fig. 4. Top port 1 of R(3,2) is currently connected to bottom port 0 of R(2,2). Divert traffic from top port 1 of R(3,2) and bottom port 0 of R(2,2). Stop top port 1 of R(3,2) and bottom port 0 of R(2,2). Disconnect top port 1 of R(3,2) and bottom port 0 of R(2,2) and move the disconnected connection to bottom port 2 of R(2,2) as shown in Fig. 7G. Start bottom port 2 of R(2,2) and top port 1 of R(3,2). Stop diverting the traffic from bottom port 2 of R(2,2) and top port 1 of R(3,2).

The port fifo is now empty. The process continues by loading the port fifo with {top port 0 of R(3,1), bottom port 0 of R(2,2)}.

The process continues by selecting top port 0 of R(3,1) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Top port 0 of R(3,1) should be connected to bottom port 2 of R(2,0) according to Fig. 4. Bottom port 2 of R(2,0) is currently connected to top port 0 of R(3,2). Divert traffic from bottom port 2 of R(2,0) and top port 0 of R(3,2). Stop bottom port 2 of R(2,0) and top port 0 of R(3,2). Disconnect bottom port 2 of R(2,0) and top port 0 of R(3,2) and move the disconnected connection to top port 0 of R(3,1) as shown in Fig. 7H. Start top port 0 of R(3,1) and bottom port 2 of R(2,0). Stop diverting the traffic from top port 0 of R(3,1) and bottom port 2 of R(2,0).

The port fifo now contains {bottom port 0 of R(2,2)}. The process continues by selecting bottom port 0 of R(2,2) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Bottom port 0 of R(2,2) should be connected to top port 1 of R(3,0) according to Fig. 4. Top port 1 of R(3,0) is currently connected to bottom port 1 of R(2,1). Divert traffic from top port 1 of R(3,0) and bottom port 1 of R(2,1). Stop top port 1 of R(3,0) and bottom port 1 of R(2,1). Disconnect top port 1 of R(3,0) and bottom port 1 of R(2,1) and move the disconnected connection to bottom port 0 of R(2,2) as shown in Fig. 7I. Start bottom port 0 of R(2,2) and top port 1 of R(3,0). Stop diverting the traffic from bottom port 0 of R(2,2) and top port 1 of R(3,0).

The port fifo is now empty. The process continues by loading the port fifo with {top port 0 of R(3,2), bottom port 1 of R(2,1)}.

The process continues by selecting top port 0 of R(3,2) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Top port 0 of R(3,2) should be connected to bottom port 0 of R(2,1) according to Fig. 4. Bottom port 0 of R(2,1) is currently connected to top port 0 of R(3,3). Divert traffic from bottom port 0 of R(2,1) and top port 0 of R(3,3). Stop bottom port 0 of R(2,1) and top port 0 of R(3,3). Disconnect bottom port 0 of R(2,1) and top port 0 of R(3,3) and move the disconnected connection to top port 0 of R(3,2) as shown in Fig. 7J. Start top port 0 of R(3,2) and bottom port 0 of R(2,1). Stop diverting the traffic from top port 0 of R(3,2) and bottom port 0 of R(2,1).

The port fifo now contains {bottom port 1 of R(2,1)}. The process continues by selecting bottom port 1 of R(2,1) because no port top or bottom has a corresponding port that is not connected and, it is the first port in the port fifo. Coincidentally, it is also the remaining port left in the port fifo. Bottom port 1 of R(2,1) should be connected to top port 0 of R(3,3) according to Fig. 4. Top port 0 of R(3,3) is currently connected to bottom port 1 of R(2,1). Divert traffic from top port 0 of R(3,3) and bottom port 1 of R(2,1). Stop top port 0 of R(3,3) and bottom port 1 of R(2,1). Disconnect top port 0 of R(3,3) and bottom port 1 of R(2,1) and move the disconnected connection to bottom port 1 of R(2,1) as shown in Fig. 7K. Start bottom port 1 of R(2,1) and top port 0 of R(3,3). Stop diverting the traffic from bottom port 1 of R(2,1) and top port 0 of R(3,3).

This completes the rewiring phase of the width upgrade of the network shown in Fig. 2 in accordance with the insertion positions shown in Fig. 3B all that remains to complete the upgrade is to connect to external sources (if desired) and activate external connections denoted by 45 and 46.

In a more sophisticated system, connections to ports on the same routers may be exchanged logically rather than physically. An example of such a swap is illustrated when interconnection network, 47, of Fig. 3B is transformed into the interconnection network shown in Fig. 8A by sapping bottom port 2 and bottom port 3 of router R(1,0). This may be accomplished logically by permanently diverting the traffic intended for bottom port 2 of router R(1,0) to bottom port 3 of router R(1,0) and the traffic intended for bottom port 3 of router R(1,0) to bottom port 2 of router R(1,0). Even if such a logical exchange is not possible the ports may be in such physical proximity to each other that a physical swap can be made while minimizing the time which the disconnections occur resulting in very little impact on throughput bandwidth. In addition, only port exchanges are used here, but natural extensions to general port permutations may also be employed.

Rather than show a complete upgrade employing the relabel, for the entire network. The upgrade and rewiring of interconnection network 47 of Fig. 3B (which is the same as interconnection networks 42, 43, 44 of Fig. 3A.) is shown in great detail here. The detail algorithm is given in Fig. 13 and flowcharted Fig. 18.

The process begins by scanning bottom ports from left to right, R(0,1) is connected to R(1,1), but according to Fig. 4, bottom port 2 should be connected to R(1,1) instead of bottom port 1. The connection to those ports are swapped as shown in Fig. 8A.

The process continues by scanning bottom ports from left to right, R(0,2) is connected to R(1,3), but according to Fig. 4, bottom port 1 should be connected to R(1,3) instead of bottom port 0. The connection to those ports are swapped as shown in Fig. 8B.

The process continues by scanning bottom ports from left to right, R(0,2) is connected to R(1,0), but according to Fig. 4, bottom port 2 should be connected to R(1,0) instead of bottom port 0. The connection to those ports are swapped as shown in Fig. 8C.

The process continues by scanning bottom ports from left to right, R(0,3) is connected to R(1,3), but according to Fig. 4, bottom port 2 should be connected to R(1,3) instead of bottom port 0. The connection to those ports are swapped as shown in Fig. 8D.

After finishing scanning the bottom ports from left to right, top ports are scanned from right to left. All top ports that are connected to the correct router are connected from the correct top port to the correct bottom port. Hence, the relabeling process is finished. The process continues by employing the same "select_port" algorithm as in Fig. 12B as before.

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, bottom port 1 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,3). Top port 2 of R(1,3) is currently connected to bottom port 0 of R(0,3). The breaking of this connection does not leave router R(0,3) or router R(1,3) with more than one broken connection. This completes the selection process for this step.

With bottom port 1 of R(0,N) selected and recalling that its corresponding port, top port 2 of R(1,3), is connected to bottom port 0 of R(0,3), the process continues by diverting traffic from

top port 2 of R(1,3) and bottom port 0 of R(0,3); stopping top port 2 of R(1,3) and bottom port 0 of R(0,3); disconnecting top port 2 of R(1,3) and bottom port 0 of R(0,3) and moving the disconnected connection to bottom port 1 of R(0,N) as shown in Fig. 8E; starting bottom port 1 of R(0,N) and top port 2 of R(1,3); and stop diverting the traffic from bottom port 1 of R(0,N) and top port 2 of R(1,3).

The process continues by selecting top port 1 of R(1,N) because scanning from right to left, it is the first port bottom or top not connected to anything and whose corresponding port (according to Fig. 4) is also not connected to anything; establishing a new connection between top port 1 of R(1,N) and bottom port 0 of R(0,3) as shown in Fig. 8F; starting top port 1 of R(1,N) and bottom port 0 of R(0,3); and stop diverting the traffic from top port 1 of R(1,N) and bottom port 0 of R(0,3).

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, bottom port 0 of R(0,N) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,2). Top port 2 of R(1,2) is currently connected to bottom port 1 of R(0,3). The breaking of this connection does not leave router R(0,3) or router R(1,2) with more than one broken connection. This completes the selection process for this step.

With bottom port 0 of R(0,N) selected and recalling that its corresponding port, top port 2 of R(1,2), is connected to bottom port 1 of R(0,3), the process continues by diverting traffic from top port 2 of R(1,2) and bottom port 1 of R(0,3); stopping top port 2 of R(1,2) and bottom port 1 of R(0,3); disconnecting top port 2 of R(1,2) and bottom port 1 of R(0,3) and moving the disconnected connection to bottom port 0 of R(0,N) as shown in Fig. 8G; starting bottom port 0 of R(0,N) and top port 2 of R(1,2); and stop diverting the traffic from bottom port 0 of R(0,N) and top port 2 of R(1,2).

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, top port 0 of R(1,N) is not connected to anything with corresponding port according to Fig. 4, bottom port 1 of R(0,1). Bottom port 1 of R(0,1) is currently connected to top port 1 of R(1,1). The breaking of this connection does not leave router R(0,1) or router R(1,1) with more than one broken connection. This completes the selection process for this step.

With top port 0 of R(1,N) selected and recalling that its corresponding port, bottom port 1 of R(0,1), is connected to top port 1 of R(1,1), the process continues by diverting traffic from bottom port 1 of R(0,1) and top port 1 of R(1,1); stopping bottom port 1 of R(0,1) and top port 1 of R(1,1); disconnecting bottom port 1 of R(0,1) and top port 1 of R(1,1) and moving the disconnected connection to top port 0 of R(1,N) as shown in Fig. 8H; starting top port 0 of R(1,N) and bottom port 1 of R(0,1); and stop diverting the traffic from top port 0 of R(1,N) and bottom port 1 of R(0,1).

Scanning from right to left, no port, top or bottom, has a corresponding port which is not connected. Again, scanning from right to left, bottom port 1 of R(0,3) is not connected to anything with corresponding port according to Fig. 4, top port 2 of R(1,0). Top port 2 of R(1,0) is currently connected to bottom port 0 of R(0,2). The breaking of this connection does not leave

router R(0,2) or router R(1,0) with more than one broken connection. This completes the selection process for this step.

With bottom port 1 of R(0,3) selected and recalling that its corresponding port, top port 2 of R(1,0), is connected to bottom port 0 of R(0,2), the process continues by diverting traffic from top port 2 of R(1,0) and bottom port 0 of R(0,2); stopping top port 2 of R(1,0) and bottom port 0 of R(0,2); disconnecting top port 2 of R(1,0) and bottom port 0 of R(0,2) and moving the disconnected connection to bottom port 1 of R(0,3) as shown in Fig. 8I; starting bottom port 1 of R(0,3) and top port 2 of R(1,0); and stop diverting the traffic from bottom port 1 of R(0,3) and top port 2 of R(1,0).

The process continues by selecting bottom port 0 of R(0,2) because scanning from right to left, it is the first port bottom or top not connected to anything and whose corresponding port (according to Fig. 4) is also not connected to anything; establishing a new connection between bottom port 0 of R(0,2) and top port 1 of R(1,1) as shown in Fig. 8J; starting bottom port 0 of R(0,2) and top port 1 of R(1,1); and stop diverting the traffic from bottom port 0 of R(0,2) and top port 1 of R(1,1).

This completes the relabel and rewire phase applied to interconnection network, 47, of Fig. 3B. In a complete upgrade, this algorithm could first be applied to interconnection network, 48, of Fig. 3B, then interconnection network, 47, of Fig. 3B, and finally interconnection network, 49, of Fig. 3B in accordance with the first "row_select" algorithm of Fig. 11. Again to complete the upgrade external connections, 45 and 46 should be connected to external sources and activated.

Though the example focused in this application is illustrated in Fig. 3B, it has been found that the best upgrade algorithm found employs the first "insertion_position" algorithm of Fig. 10 and shown in Fig. 3A; uses the row order dictated by the first "row_select" algorithm of Fig. 11; and on each interconnection network selected by the above row order, optionally uses the "relabel" algorithm given in Fig 13, and the "select_port" algorithm of Fig. 12B.

In the two other examples depicted in Fig. 3A and Fig. 3C, after the rewire (and relabel) phase using the components of choice. The upgrade is completed by connecting and activating external connections, 40 and 41, for the example in Fig. 3A or 50 and 51, for the example in Fig. 3C.

The process depicted in application may be used also to reduce the width of the network. Simply reverse the process shown here. In summary, the process would involve disconnecting external connections to routers to be removed; rewiring each interconnection network in the same manner as those described above; and finally removing the unwired routers.

Though this invention is applied to a RBCCG network composed of rows of routers of consistent number per row and fanout per router. It may be applied to fault tolerant networks composed of rows of routers of varying number per row and fanout per router where the rows are connected with interconnection networks of the same type as that used in a RBCCG network. It also applies to any redundant multistage network for which a width augmentation is defined. In addition, it is not restricted to routers and may be applied to networks composed of other types of switching elements.

Although the present invention has been described above in terms of specific embodiments, it is anticipated that alteration and modifications thereof will no doubt become apparent to those skilled in the art. It is therefore intended that the following claims be interpreted as covering all such alterations and modifications as falling within the true spirit and scope of the invention.

2024-04-24